Music Genre Classification Using Convolutional Neural Network

Thesis

Submitted to the

G. B. Pant University of Agriculture & Technology
Pantnagar- 263145, Uttarakhand, India

By

NEEMA BHANDARI

B. Tech.
(Information Technology)

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master of Technology

(COMPUTER ENGINEERING)

February, 2021

ACKNOWLEDGEMENT

Pursing M. Tech research is just like climbing a high peak, step by step, accompanied with hardships, encouragement and trust and with so many people's kinds help. Firstly, I would like to express my deepest sense of gratitude to the almighty "GOD" foe blessing me with enough patience, endurance & strength to overcome all the hurdles that crossed my path. Accomplishment of this thesis is the result of benevolence of omnipotent almighty and blessing of my teachers. I am overwhelmed with joy to evince my profound sense of reverence and gratitude to ***Prof. B.K. Singh*** *Associate Professor and Chairman of my Advisory Committee, for his selfless guidance and continuous encouragement through the tenure of my investigation and preparation of dissertation.*

The author expresses his deep sense of gratitude to members of his advisory committee ***Prof. Jalaj Sharma****, Associate professor, Department of Computer Engineering, and* ***Prof. P. K. Mishra****, Assistant Professor, Department of Computer Engineering for giving me valuable suggestions and help during the thesis work.*

It is privilege to express my heartiest regards & sincere thanks to all the faculty members ***Dr, S. D. Samantaray****,* ***Dr. Rajiv Singh****,* ***Prof. Sunita Jalal*** *and* ***Prof. Chetan Singh Negi*** *for their cooperation and support throughout the degree programme.*

My sincere thanks to all my Librarian, Director, Dean, College of Technology, Dean, College of Post Graduate Studies and Registrar for providing me the essential facilities to conduct the proposed investigation.

I would like to express my endless love, thanks to my father ***Mr. Ram Singh Bhandari****, mother* ***Kamla Bhandari*** *and my husband* ***Mr. Pankaj Singh*** *for their love, encouragement and care.*

Pantnagar
February, 2021

(Neema Bhandari)
Author

CERTIFICATE

This is to certify that the thesis entitled **"Music Genre Classification using Convolutional Neural Network"** submitted in partial fulfillment of the requirements for the degree of **Master of Technology** with major in **Computer Engineering**, of the College of Post-Graduate Studies, G. B. Pant University of Agriculture and Technology, Pantnagar, is a record of bona fide research carried out by **Ms. Neema Bhandari,** Id. No. **54100** under my supervision and no part of the thesis has been submitted for any other degree or diploma.

The assistance and help received during the course of this investigation have been acknowledged.

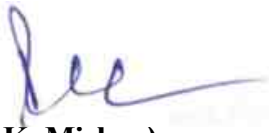
Pantnagar
February, 2021

(B K Singh)
Chairman
Advisory Committee

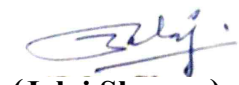
CERTIFICATE

We, the undersigned, members of the Advisory Committee of **Ms. Neema Bhandari,** Id. No. **54100,** a candidate for the degree of Master of Technology with major in **Computer Engineering,** agree that the thesis entitled **"Music Genre Classification using Convolutional Neural Network"** may be submitted in partial fulfillment of the requirements for the degree.

Chairman
Advisory committee



(P. K. Mishra)
Member



(Jalaj Sharma)
Member

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Full Form
CNN	Convolutional Neural Network
MFCCs	Mel frequency cepstral coefficients
RMS	Root Mean Square
MGC	Music Genre Classification
STFT	Short Time Fourier Transform
DFT	Discrete Fourier transform
DCT	Discrete Cosine Transform
FC	Fully Connected Layer
MPEG	Moving Picture Experts Group
BN	Batch Normalization
FT	Fourier Transform
IFT	Inverse Fourier Transform
FFT	Fast Fourier Transform
2D	Two-Dimensional
PS	Power Spectrum
DC	Delta Coefficients

Introduction

Chapter-1 INTRODUCTION

Music has been a part of our lives for a long time. Music has the power of spawning the emotions in humans' heart that are impossible otherwise. However, music is complete in itself, it is further subdivided into genres. In this chapter we will discuss Indian music, their Genre, and Overview of Music Genre classification using convolutional neural networks.

1.1 Overview

About the source of music, it was speculated by Charles Darwin that modern-day languages had its roots in music. Darwin proposed that during the courtship our ancestors used musical tone and rhythms and from this "primal music" speech was developed (cf Sacks, 2007, p. xi). Storr (1993) opines that music had its origins in the "prosodic exchanges between the mother and the infant" (p. 23). Later it developed into a form of communication between adults. As time went by, speech developed and replaced music as the medium of conveying information. Herbert Spencer says that music originated from emotional speech. William James believed that music had an "accidental genesis", an output of possessing a "hearing organ" (Sacks, 2007, p. xi). These different views put together generate the notion that music in its intelligible form, can be attributed to the human species. Humans not only comprehend music, but they can also compose it. We might find the sounds produced by other species musical, but these sounds are considered as biological functions of those animals.

The Indian assumption of music can be traced back to the Vedas which laid great stress on the audible sound. Sound in the Vedic tradition is the manifestation of the Cosmic sound. The audible sound is called *ahata* and the cosmic sound is called *anahata.* While everyone can hear the audible sound by their ears, the cosmic sound or *anahata* can be heard only by the *yogi* through deep contemplation. Music consists of the *ahata* and hence is believed to be a manifestation of the *anahata*. Hence as per the Hindu tradition, music contains the power, energy, and consciousness of the divine. Similar to the Chinese, the Hindu tradition believes that sound or music possesses the power to influence the emotions and the human's mind and further it can shape and change the physical actions of the world. Music was seen as created by humans, coupled with voice and the usage of musical gadgets, and hence it was considered to have very powerful effects. Music was considered to be attractive even to the deity (Tame, 1984). Hence the place of music in religion has been central in the Indian custom since ancient times. (Marcus, 2003).

1.2 Motivation

A single music genre is a set of different features that is combined with a specific pattern of rhythm, melody, harmony, instruments, mood and attitude, lyrics and language. In the western music there has been much work done in the area of automatic tagging, genre recognition, classification and comparative studies as compared to the indian music. Classical indian music is divided into two-parts namely Hindustani classical and Carnatic classical. Hindustani classical music is prevalent in north and central India and Carnatic classical is prevalent in south India. Hindustani classical music has Dhrupad, Khayal, Tarana, Dhamar and Sadra as classical genre and Thumri, Dadra, Tappa, kajari, Bhajan, Natyageet, Qawwali and Ghazal as Semi Classical Genre. Music has multiple folk genres based on region, film music and modern music which are influenced by multiple Indian and western genres. Folk music is based on tonal element and classical and semi classical music is based on Raga(s) which describes the emotion, mood or sentimental expression in a microtonal scale form. Almost all indian music uses a single note played which is given in specific order but western music has chord which are played simultaneously. Svar or Tone is the basic unit of melodic structure in the indian music. There are seven common Svar in indian music known as Shadi(Sa), Rishabh(Re), Gandhar(Ga), Madhyam(Ma), Pancham(Pa), Dhaivat(Dha) and Nishad(Ni). Indian music genres are different from other music genres because of its own well defined structure.

1.3 Indian Music Genres

A combination of classical music and folk music is called Semi-Classical Indian music. These genres have some specific patterns like classical music but it is influenced by a religious area. A semi-classical music genre is used for optimizing the time of the users and it helps to allocate a large number of tracks in a different category of the music. Indian music contains a different genre, such as mainstream music and art music, or religious music and secular music. Genre helps the user to easily shortlist the music of the user's interest. Music industry is vastly spread and shortlisting music track according to user’s preference is difficult. The task is to classify various properties of provided music recordings, based on extracted sound characteristics. There are several phases of evolution and transformation of semi classical music during the course of time. The main instrument of semiclassical is Samvadini, Santoor, Sarangi, Sarod, Sitar, Tabla and Veena, etc. these instruments are the type of sting, wind, impact, and percussion. It is a challenging problem to capture the inherent structure of

Indian music and use it for genre classification. We have included six semi-classical music genres in our work.

1.4 Music Genre Classification

Music genre classification has been gaining attention with the rise of digital music and it is a useful tool for semantic information of music tracks in offline and online music collections. A music genre refers to a specific class of music with a set of common properties. A mere perception of the music of that class can help one to distinguish it from other classes. A music genre classification systems first extract information about each track that is useful for distinguishing the genre, this is known as Feature.

Ragas are the central structure of hindustani classical music. We have tried to identify eighteen ragas played by the three chordophones by signal processing approaches. Here we have used our own locally generated database. We have addressed the problem of Instrument Identification and raga recognition as two separate tasks. The literature of musical devices or instruments for the classification is split into two main approaches: monophonic classification, the identity of musical devices playing solo; and polyphonic classification, where musical instruments are identified while playing in an ensemble; of which we have used the first one.

The basic problem in sound source (musical instrument) recognition is contextual variation. Sound waves formed by a certain source are differently produced at each event. If they were similar, then the recognition could take place simply by comparing the waves into some characteristic templates stored into memory. In the real world, the waves produced at different times are very different. This is due to the physical process generating the sound is very seldom exactly the same at different times. In addition, the position of a source with respect to a listener, and the acoustic features of the environment affect the sound waves.

A lot of Researchers have provided an overview of the music genre classification techniques that had been tried to date, including features and classifiers used with a comprehensive overview of deep learning algorithms suitable for application in this classification problem. Most of the researchers have used a large number of features extracted from the audio and they include spectral features, temporal features, MPEG-7 descriptors, MFCC coefficients, and their derivatives. They have used these features for

training various classifiers and used these trained classifiers to test the remaining signals for instrument classification. Classification accuracy was reported as result of classification. In most of the cases classification accuracy is between 70 to 98%.

1.5 Feature Exaction

Feature Exaction is a way to capture the information from a track that is similar for tracks in the same genre and different for songs. Music information retrieval (MIR) is extracting higher-level information such as genre, artist, chord, scale, or instrumentation from music. This Information properly represents the musical characteristics of each song. The main dimension of the music feature is melody, rhythm, timber and spatial feature, etc. Some of the features are as follows.

- **Tone** is defined as any sound of a definite pitch. The term tone is often used interchangeably with 'note'. A note is referred to what is seen written on a musical score to denote a tone. Tonality features are chromagram pitch class energies, cross-correlation of the chromagram, dimensional tonal, and centroid vector from the chromagram.
- **Pitch** is used to identify the highness and the lowness of a musical tone. A variance of pitch is called Pitch features.
- **Tempo**: The overall speed of the music is performed is called Tempo.
- **Rhythm**: It refers to the movement of musical tones regarding time and the way they group together into units. Rhythm features are mean and variance of notes onset time, average frequency of events, tempo, strength of beats (pulse clarity).
- **Timbre features** are mean and variance of attack time of notes onset, roll off frequency, brightness, 13 Mel-Frequency Cepstral Coefficients (MFCCs), sensory dissonance, irregularity.
- **Spectral-shape features** is zero-crossing rate, spread, centroid, skewness, kurtosis, mean and variance of Inverse Fourier Transform of logarithm of spectrum, flatness, mean and variance of spectrum, mean and variance of spectral flux, mean and variance of spectrum peaks.
- **Energy features** are mean and variance of energy envelope, RMS, percentage of frames having less than average energy.

1.6 Short time Fourier transform

For the timbral texture feature the short time Fourier transform is very essential because it is an analysis of the frequency band of the music. Some frequency features are calculated on the behalf of the STFT and these are spectral centroid, most prevalent frequencies, spectral rolloff, spectral flux, the frequency spectrum and time domain zero crossing.

1.6.1 Mel-Frequency Cepstral Coefficients

MFCCs previously used in speech recognition. The word Mel scale refer to the human auditory model and it also refers to the word melody to indicate that the scale is based on pitch comparisons. It represents a set of short-term power spectrum characteristics of the sound.

1.6.2 Zero Crossing Rate

The rate of sign changes of signal is called zero crossing rate. It is helpful for audio processing and music information retrieval. For extremely percussive sounds zero crossing rates have a higher value.

1.6.3 Spectral Centroid

Spectral centroid is indicating the "centre of mass" of a sound. it is located and calculated by the weighted mean of the frequency present in the sound.

1.6.4 Spectral Rolloff

It is a measure of the shape of the signal. It represents the frequency which is below a particular percentage of the total spectral energy.

1.6.5 Chroma Frequencies

Chroma features is a powerful representation for music where the entire spectrum is projected onto 12 bins and 12 different semitones or chroma of the musical octave.

1.7 Classification of Music Genre

For the Classification of music genre, we have used convolution neural networks (CNN). Basically, CNN uses three layers for classification.

1.7.1 Convolution Layer

The first layer is known as convolution layer and it is used for extracting features from an input image. It preserves the connection between pixels by learning image features by using small squares of input data. Convolution layer perform mathematical operations. This mathematical operation takes two inputs. One input is an image matrix and the second input is a filter or kernel.

1.7.2 Pooling Layer

Pooling layers is used to reduce the number of parameters for the large image. One of the most important terms is spatial pooling. Spatial pooling is used for sub sampling or down sampling which is very helpful for reducing the dimensionality of each map and retains important information.

1.7.3 Fully Connected (FC) Layer

Flattening layers work in between the convolutional layer and the FC layer. It is used to transform a two-dimensional matrix of features into a vector that can be fed into a vector that can be fed into a fully connected neural network classifier.

Like a neural network, matrix is compressed into vectors and feeded to a FC layer. A FC layer also known as the dense layer. It is the results of the convolutional layers that are fed through one or more neural layers to generate a prediction.

1.8 Objectives

- The Objective of this work is to extract the feature of semi classical genre by using the classification technique ‘convolutional neural network (CNN)’.
- Proposed work will use Indian semi classical music to extract features and classify its genre.
- Propose and design a method that is easy to implement.
- Compare the accuracy of indian and western music genres.

1.9 Thesis Organization

In **Chapter-1,** Overview of Music genre, background of Indian music, Music features, feature extraction techniques, convolutional neural network, layer of convolution neural network are studied.

Chapter -2, review of literature about earlier reported work related to Music Genre Classification and Convolution Neural Network have been discussed.

Chapter -3, the materials and methods are discussed for carrying out this research work, it also contains the steps which are used for the designing of the proposed work.

Chapter -4, The proposed work uses the Python platform for generating the model, to analyze the Model performance for Music Feature extraction and classification accuracy.

Chapter – 5, Conclusion and Future Scopes of proposed work have been discussed.

Review of Literature

Chapter-2 REVIEW OF LITERATURE

A lot of research has been done in music genre classification using machine learning. This chapter presents a brief review of the research done by various researchers in the field of machine learning.

Tzanetakis, G. and Cook, P. 2002 proposed automatic classification of audio signals into a hierarchy of musical genres. They characterized the music genre by the common characteristics of the instrument, rhythm structure, and harmonic content of the music. They proposed a framework for developing and evaluating the content-based analysis of musical signals.

Song, Y. and Zhang, C., 2008 proposed Content-Based Information Fusion for Semi-Supervised Music Genre. They proposed an information fusion framework for the semi-supervised distance-based music genre classification problem. They used a regularized least-square framework as the basic classifier, which only involves the similarity scores among different music tracks. They presented a similarity score that multiplies different scores based on different distance measures. Particularly the distance measures are not restricted to the Euclidean distance. By adding a weight to each single distance-based score, they propose an expectation-maximization (EM) algorithm to adaptively learn the fusion scores.

Kao, M. *et al.* 2009 proposed Tempo and beat tracking for audio signals with music genre classification. They believed that most people follow the music to hum or the rhythm to tap sometimes. People may get different meanings of a music style if it is explained or felt by different people. Therefore, people cannot obtain a very explicit answer if there is no music notation. Tempo and beats are very important elements in perceptual music. Therefore, tempo estimation and beat tracking are fundamental techniques in automatic audio processing, which are crucial to multimedia applications. They first developed an artificial neural network to classify the music excerpts into the evaluation preference. And then, with the preference classification, they obtained an accurate estimation for tempo and beats, by either Ellis's method or Dixon's method. They tested their method with a mixed data set which contained ten music genres from the 'ballroom dancer' database. Their experimental results showed that the accuracy of their method is higher than only one individual Ellis's method or Dixon's method.

Anglade, A. *et al.* 2010 proposed Improving Music Genre Classification Using Automatically Induced Harmony Rules. They presented a new genre classification framework using both low-level signal-based features and high-level harmony features. A state-of-the-art statistical genre classifier based on timbral features was extended using a first-order random forest containing for each genre rules derived from harmony or chord sequences. This random forest has been automatically induced, using the first-order logic induction algorithm TILDE, from a dataset, in which for each chord the degree and chord category are identified, and covering classical, jazz and pop genre classes. The audio descriptor-based genre classifier contains 206 features, covering spectral, temporal, energy, and pitch characteristics of the audio signal. The fusion of the harmony-based classifier with the extracted feature vectors is tested on three-genre subsets of the GTZAN and ISMIR04 datasets, which contain 300 and 448 recordings, respectively. Machine learning classifiers were tested using 56 5-fold cross-validation and feature selection. Results indicated that the proposed harmony-based rules combined with the timbral descriptor-based genre classification system lead to improved genre classification rates.

Marques, G. *et al.* 2011 proposed Short-term Feature Space and Music Genre Classification. They recognized that in music genre classification, most approaches rely on statistical characteristics of low-level features computed on short audio frames. In these methods, it is implicitly considered that frames carry equally relevant information loads and that either individual frames, or distributions thereof, somehow capture the specificities of each genre. The research of the representation space defined by short-term audio features with respect to class boundaries, and compare different processing techniques to partition this space. These partitions were evaluated in terms of accuracy on two genre classification tasks, with several types of classifiers. Their Experiments showed that a randomized and unsupervised partition of the space, used in conjunction with a Markov Model classifier lead to accuracies comparable to the state of the art. They also showed that unsupervised partitions of the space tend to create less hubs.

Sanden, C. *et al.* 2012 proposed A Perceptual Study on Music Segmentation and Genre Classification. MIR is an interdisciplinary area that is engaged in the retrieval of information from music. It includes various tasks, such as music classification, clustering, perception and cognition, etc. They presented perceptual studies on segmentation and genre classification, two indispensable steps in the MIR process. Segmentation attempts to

capture ‘drastic’ changes in music and provides a basis for further perceptual and computational analysis while genre classification amounts to separating music into different groups such that each group uniformly represents a music genre. their perceptual study considers various related issues. The goal of their work is to explore and deepen the understanding of the relationship between perceptual surface and perceptual structure of music through segmentation by human subjects and reveal and demonstrate the multi-label nature of genre classification.

Li, W. *et al.* 2013 proposed Music content authentication based on beat segmentation and fuzzy classification. Digital audio has been ubiquitous over the past decade. Since it can be easily modified by editing tools, there has been a strong need to protect its content for secure multimedia applications. Previous audio authentication algorithms are mainly focused on either human speech or general audio with music as part of the test data, while special research on music authentication has been somewhat neglected. They propose a novel algorithm to protect the integrity and authenticity of music signals. Their research includes the following: Music is segmented into beat-based frames, which not only endows the authentication units with more semantic meaning but also perfectly resolves the challenging synchronization problem and secondly Robust hashes are generated from chroma-based mid-level audio feature which can appropriately characterize the music content and integrated with an encryption procedure to ensure the security against malicious block-wise vector quantization attack and Fuzzy logic is adopted to make the authentication decision in the light of three measures defined on bit errors, coinciding with the inherent blurred nature of authentication. The experiments exhibit good discriminative ability between admissible and malicious operations.

Ntalampiras, S. 2014 proposed Directed Acyclic Graphs for Content Based Sound, Musical Genre, and Speech Emotion Classification in Journal of New Music Research. His work introduces the methodology of Decision Directed Acyclic Graphs (DDAG) to the scientific domain of content-based audio signal processing. He applied the particular methodology to three multiclass classification problems involving the categories of generalized sound events, musical genres, and speech expressing emotional states. A decision graph is constructed which breaks the overall problem into a series of two-class ones. The order of the graph nodes is revealed using a clustering criterion based on the Kullback-Leibler divergence. Every graph node is composed by two hidden Markov

models, each one representing the class which participates in the specific problem. He extracted three heterogeneous feature sets (Mel-Filter bank, MPEG-7 Audio Spectrum Projection and Perceptual Wavelet Packets) out of each recording and fused them for training the HMMs. Extensive comparative experiments are conducted using the following three datasets: (a) a combination of professional sound effects collections, (b) GTZAN musical genre database, and (c) BERLIN emotional speech corpus. The results demonstrate the superiority of the DDAG classification approach over the standard HMM approach regardless the application task

Betsy, S. and Bhalke, D. G. 2015 proposed Indian Tamil Music genre classification using MFCCs and timbral features. They divided the music genre category by using harmonic content, rhythmic structure, and instrumentation. They build the classifier model by using K-Nearest Neighbour (KNN) and Support Vector Machine (SVM).

Yoon, J. *et al.* 2016 proposed music genre classification using Feature subset search. They suggested a new method for selecting genre-discriminative feature subset from a large number of musical features. They showed that the proposed method is able to improve the genre recognition accuracy compared to the traditional selection method.

Viswanathan, A. 2016 proposed Music Genre Classification. The purpose of his research was to build a machine learning model to predict the genre of a song. His work analyzes a song as a wave and determines the factors involved and predicts the genre. Application of the model is primarily to automate the process of genre identification which shall aid in building of music recommendation systems.

Mutiara, A. *et al.* 2016 proposed Musical Genre Classification Using SVM and Audio Features. The research on advanced MIR increases as well as a huge amount of digital music file distribution on the internet. Musical genres are the main top-level descriptors used to organize digital music files. Most of the work in the labeling genre is done manually. Thus, an automatic way of labeling a genre to digital music files is needed. They understand that the most standard approach to do automatic musical genre classification is feature extraction followed by supervised machine-learning. Their research aims to find the best combination of audio features using several kernels of non-linear Support Vector Machines (SVM). The 31 different combinations of proposed audio features are dissimilar compared in any other related research. Furthermore,

among the proposed audio features, Linear Predictive Coefficients (LPC) have not been used in other works related to musical genre classification. LPC was originally used for speech coding. Experimentation in classifying digital music files into a genre is carried out. The experiments are done by extracting feature sets related to timbre, rhythm, tonality, and LPC from music files. All possible combinations of the extracted features are classified using three different kernels of SVM classifier that are Radial Basis Function (RBF), polynomial, and sigmoid. The result shows that the most appropriate kernel for automatic musical genre classification is the polynomial kernel and the best combination of audio features is the combination of musical surface, MFFC, tonality, and LPC. It achieves 76.6 % classification accuracy.

Bhalke, D. *et al.* 2017 Proposed Automatic Genre Classification of Indian Tamil Music and Western Music Using Fractional Fourier Transform Based Mel MFCC and Timbral Features. In the Indian Tamil music genre, they included classical Carnatic music & folk music and for the western genre, they included Rock and Classical music. They compared their accuracy by the feature combination of Spectral Roll-off, Spectral Flux, Spectral Skewness, and Spectral Kurtosis, combined with Fractional MFCC features.

Samad, A. *et al.* 2018. Proposed Traditional Malay Music Genre Classification of Using Spectrogram Correlation. They focused on the three distinct parts first is to construct spectrogram which retains the most salient feature of music and the second one is to construct the template which helps to account the variation in music as well as the music progresses. The third one is the template matching which is based on spectrogram image cross-correlation. Their experiments were done with seven traditional Malay music genres and their recognition accuracy is dependent on the number of the segment. The number of segments used to construct the filter template which is related to the length of the music segment. Their recognition rate of 61.8 percent was obtained for music segments lasting 180 seconds using six relatively short excerpts.

Meenakshi, K. *et al.* 2020 proposed Music Genre Classification using Lyric Mining. They considered the need for such classification, research works are being carried out to develop methodologies that can distinguish the music based on individual mood. The research on the traditional method of audio feature analysis proposes to develop a system that analyses the lyrics dataset of the songs based on the features extracted from the training phase and they can predict the mood of the song that is presented to the system at

the validation stage. Their proposed system is considering five moods containing one hundred songs each, for the validation purpose. The system is capable of predicting the mood of the song based on the analysis of the lyric text.

Seth, A. 2020. Proposed genre prediction for music recommendation using machine learning. They proved that Music applications are one of the most used applications in the world. Consumers can hear the song they like but it is difficult for them to find songs from the vast number of songs list. The flow of their research is to increase the efficiency of music recommendation in terms of the genre based on the decision-tree which helps the users to get the music according to their preferences. Their model uses age and gender as an input set and genre as output. The model will predict the genre according to age and gender and the decision tree helps to reduce the complexity of the model.

Dabas, C. *et al.* 2020 proposed Machine Learning Evaluation for Music Genre Classification of Audio Signals. They see music genre classification has its own popularity index in the present times. Machine learning can play an important role in the music streaming task. Their research proposes a machine learning-based model for the classification of music genres. The evaluation of the proposed model is carried out while considering different music genres as blues, metal, pop, country, classical, disco, jazz, and hip-hop. Different audio features utilized in this study include MFCC, Delta, and temporal aspects for processing the data. The implementation of the proposed model has been done in the Python language. The results of the proposed model reveal an SVM accuracy of 95%. The proposed algorithm has been compared with existing algorithms and the proposed algorithm performs better than the existing ones in terms of accuracy.

The Literature review provides a background work that has been done in the field of music genre classification. From the review of literature, it may be observed that the area of genre classification had been the most interrogated area in the field of music. For more than 10 years researchers have struggled with the problem of music genre classification. Next chapter explains various methods that have been applied to music genre classification.

Materials and Methods

Chapter-3 MATERIALS AND METHODS

This chapter explains the materials and methods involved in the successful implementation of music genre classification using CNN.

3.1 Materials

The basic methods for classifying music genres requires following hardware and software.

3.1.1 Hardware Used

1. **CPU**: 2 x 64-bit 2.8 GHz 8.00 core i3.
2. **RAM**: 4 GB
3. **Storage**: 10 GB or above

3.1.2 Software Used

3.1.2.1 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.

1. **Anaconda Navigator:** Anaconda Navigator is a desktop GUI that comes with Anaconda Individual Edition. It makes it easy to launch applications and manage packages and environments without using command-line commands

2. **Anaconda Prompt:** Anaconda command prompt is just like command prompt, but it makes sure that you are able to use anaconda and conda commands from the prompt, without having to change directories or your path.

3.1.2.2 Spyder and PyCharm

Spyder, the Scientific Python Development Environment, is a free integrated development environment that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring and rich navigation capabilities.

3.2 Data Set collection and Feature Extraction

The main challenging task of the research is collection of data base because no predefined data set is available. We have collected songs of different genre from different commercial site's and YouTube and made it compatible with our approach. We took 30 second length of a song and 50 songs in each six different semi classical genres. In this work we have considered six semi classical Indian music genres like Bhajan, Abhang, kajari, tappa, qawwali and thumri.

3.3 Understanding of Audio data processing

There are many points we have to know for audio processing and music genre classification.

3.3.1 Reading Audio Files: After the collection of the data our prime task is to load and display the characteristics of an audio data file. There are two major components that work on audio data: one is Amplitude and second one is sampling rate or sampling frequency (number of samples per unit or second). For the visualization of the audio file we can plot it corresponding to Amplitude and time as shown in figure 3.1.

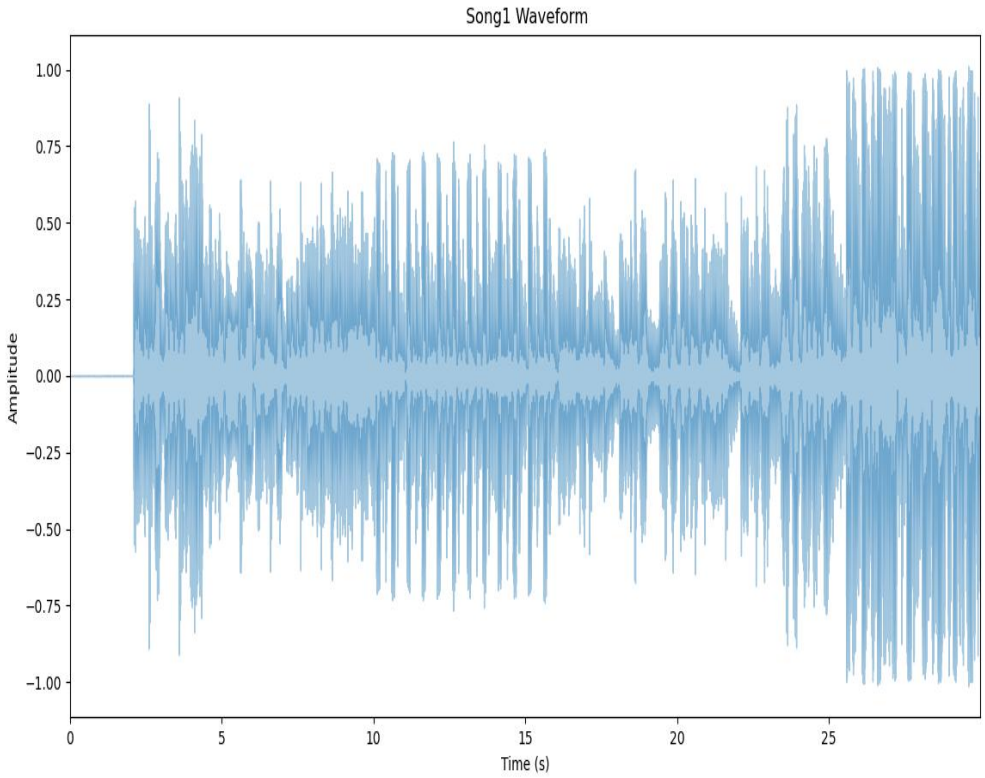


Fig. 3.1: Waveform representation of the music

The figure 3.1 shows the Time- domain representation of the audio data. It shows amplitude of sound wave changing with time and it is not very informative.

3.3.2 Fourier Transform (FT)

For the analysis of audio data, it is important to transform it into the Frequency domain. The Frequency domain represents the number of frequencies present in the audio signal. Fourier Transform is a mathematical method which converts a continuous signal from time domain to frequency domain. It also gives the magnitude of each frequency present in the audio signal. Just opposite of Fourier Transform is called Inverse Fourier Transform

3.3.3 Fast Fourier Transform (FFT)

Fast Fourier Transformation calculate the Discrete Fourier Transform (DFT) of a given Signal. It considers discrete signals as input. DFT converts discrete signals into its frequency constituents. In simple ways we can say DFT or FFT convert the time domain discrete signal into a Frequency domain.

3.3.4 Short Time Fourier Transformation

Computation of Fourier transform of partition of signal is called Short time Fourier Transformation (STFT)and it is widely used for audio applications like noise reduction, pitch detection, pitch shifting and other audio applications. STFT computes a FFT of windowed data frames. The window is 'slides' or 'Hops' forward through time. Now we derive the implementation of STFT from its mathematical definition which are expressed using following equations.

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x(n+mR)\omega(n)e^{-j\omega(n+mR)} \quad 3.1$$

$$X_m(\omega) = e^{-j\omega mR}\sum_{n=-\infty}^{\infty} x(n+mR)\omega(n)e^{-j\omega n} \quad 3.2$$

$$X_m(\omega) = e^{-j\omega mR} DTFT_w\,(SHIFT_{-mR}(x).\omega) \quad 3.3$$

Where

$x(n)$ = input signal at time n

$w(n)$ = length M window function (eg., Hamming)

$X_m(\omega)$ = DTFT of windowed data centered about time mR

R= hop Size, in samples, between successive DTFTs

In this way the data centered about time mR are translated to time 0, multiplied by the window w , and then the DTFT is performed and the non zero portion of windows data centered on time 0. Now DTFT can be replaced by DFT or FFT and it is a very effective sample of the DTFT in frequency. Sample of the frequency axis is preserving the information of the signal in the proper time limit.

Let M denote the window length and $N >= M$ be the DFT length. Then sampling at $w = w\_k = 2\pi k/N \quad k = 0,1,2, \ldots . N-1$, and using the fact that the window $w(n)$ is time-limited to less than N samples centered about time zero, yields following equations

$$X_m(\omega_k) = e^{-j\omega_k mR} \sum_{n=-N/2}^{\frac{N}{2}-1} x(n+mR)w(n)e^{-j\omega_k n} \qquad 3.4$$

$$X_m(\omega_k) = e^{-j\omega_k mR}\, DTFT_{N\omega_k}\left(SHIFT_{-mR\,(x).w}\right) \qquad 3.5$$

Since indexing in the DFT is modulo N , the sum over n can be rotated to a sum from 0 to $N-1$ as is conventionally implemented for the DFT. In practice, this means that the right half of the windowed data frame goes at the beginning of the FFT input buffer, and the left half of the windowed frame goes at the end, with zero-padding in the middle

3.3.5 Spectrogram

Spectrogram is a visual representation of frequency of given audio signal with time and its plot representation – one of axis of the plot shows the time and second axis shows frequency and the color representation is magnitude (observed frequency at particular time) as shown in figure 3.2.

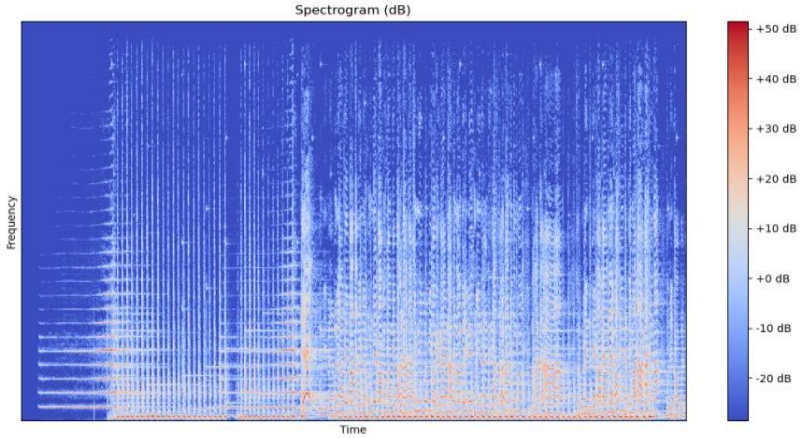


Fig3.2: Spectrogram representation of audio signal with time

3.3.6. Audio Recognition using Spectrogram Features

By Using MFCC's Feature extraction we generated a spectrogram. This spectrogram includes the 2D matrix and it represents the frequency magnitudes (MFFC coefficient) with time for a given audio signal.

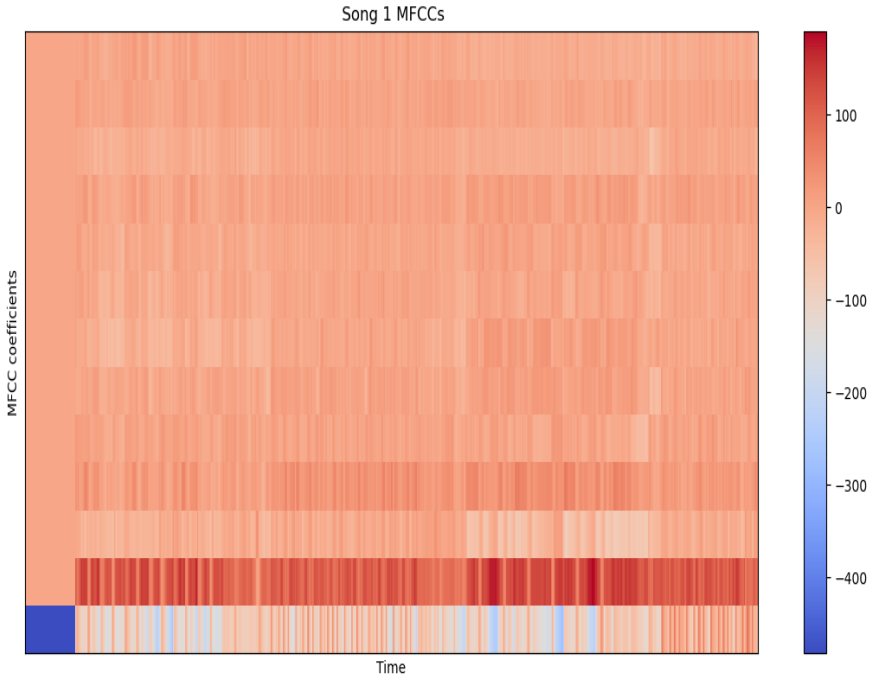


Fig 3.3: MFCCs Representation

figure 3.3 represents the audio phrases from left to right in a timly manner. This is very helpful in the classification problem. To classify the music or audio we have used the CNN technique.

3.4 Feature Extraction

The next phase of this work is extracting the features from the data set for the classification. There are several techniques used for audio feature extraction. We have used the MFCCs method for feature extraction. The MFCCs feature extraction technique basically includes windowing the signal, applying the DFT, taking the log of the magnitude, and then wrapping the frequencies on a Mel scale, followed by applying the inverse DCT. The detailed description of various steps involved in the MFCC feature extraction is explained below.

3.4.1 Implementation of MFCC

The Mel scale is referred to pitch or a pure tone which is an actual measure in frequency. Humans can understand the discerning small changes in pitch at low frequencies or at high frequencies. Including this Mel scale makes features match more relatively what humans hear.

The audio signal is a time-varying signal. For unchanging acoustic characteristics, audio needs to be examined over an appropriately short period of time. Therefore, audio analysis must always be carried out on short segments across which the audio signal is assumed to be stationary. Short-term spectral measurements are typically carried out over 20ms windows, and advanced every 10ms. Advancing the time window every 10ms enables the temporal characteristics of individual speech sounds to be tracked, and the 20ms analysis window is usually sufficient to provide good spectral resolution of these sounds, and at the same time short enough to resolve significant temporal characteristics. The purpose of the overlapping analysis is that each speech sound of the input sequence would be approximately centered at some frame. On each frame, a window is applied to taper the signal towards the frame boundaries. Divide the test audio signal into several short-wave frames to keep the audio signal constant. Generally, Hanning or Hamming windows are used. Calculating the periodogram estimates the power spectrum for each frame. This tells us the frequencies present in the short frames.

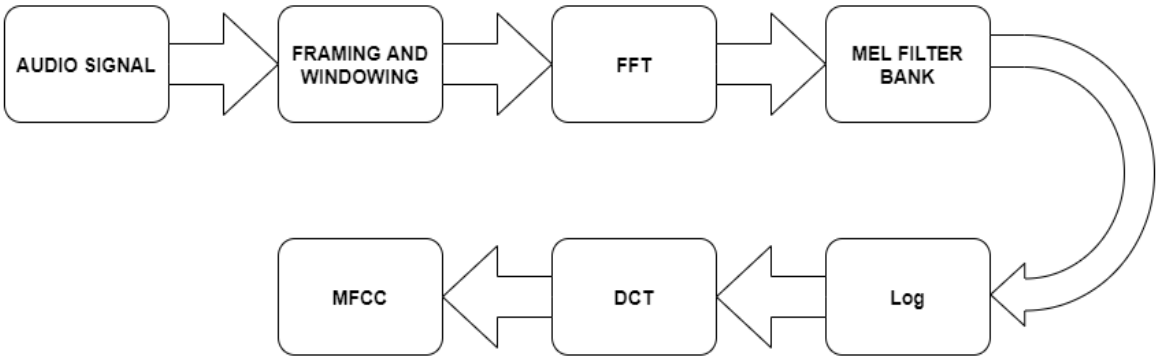


Fig3.4: Extraction of MFCCs from audio signal

Taking the power spectra into the mel filter bank and collecting the energy in each filter to sum it. We will then know the number of energies existing in the various frequency regions.

The formula for converting from frequency to Mel scale is expressed using equation 3.6

$$M(f) = 1125\ ln\ ln\ (1 + \frac{f}{700}) \qquad 3.6$$

where f denotes the physical frequency in Hz, and M denotes the perceived frequency. To go from Mel back to frequency equation 3.7 is used

$$M^{-1}\ (m) = 700\ (exp\ exp\ \left(\frac{m}{1125}\right) - 1\ ) \qquad 3.7$$

The next steps are applied to every single frame, one set of 12 MFCC coefficients is extracted for each frame. A short aside on notation: we call our time domain signal $s(n)$. Once it is framed we have $s_i(n)$ where n ranges over 1-400 (if our frames are 400 samples) and i ranges over the number of frames. When we calculate the complex DFT, we get $s_i(k)$ where i denotes the frame number corresponding to the time-domain frame. $P_i(k)$is then the power spectrum of frame i. To take the Discrete Fourier Transform of the frame, perform the following equation 3.8 is used

$$s_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N} \quad 1 \le k \le K \qquad 3.8$$

where $h(n)$ is an N sample long analysis window (e.g. hamming window), and K is the length of the DFT. The periodogram-based power spectral estimate for the speech frame $P_i(n)$ is given by equation 3.9

$$P_i(k) = \frac{1}{N}\ |\ S_i(k)^2| \qquad 3.9$$

This is called the Periodogram estimate of the power spectrum. We take the absolute value of the complex Fourier transform, and square the result. We would generally perform a 512-point FFT and keep only the first 257 coefficients.

Calculate the logarithm of the filter bank energies. To calculate filter bank energies, we multiply each filter bank with the power spectrum, then add up the coefficients. Formula for calculating filter bank is represented by equation 3.10.

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases} \qquad 3.10$$

where m is the number of filters we want, and $f()$ is the list of $M + 2$ Mel spaced frequencies.

Calculate the Discrete Cosine Transform (DCT) of the result. The DCT applied to the transformed Mel frequency coefficients produces a set of cepstral coefficients. Prior to computing DCT, the Mel spectrum is usually represented on a log scale. This results in a signal in the cepstral domain with a frequency peak corresponding to the pitch of the signal and a number of formants representing low frequency peaks. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring or truncating higher order DCT components. Keep the first 13 DCT coefficients. Remove the higher DCT coefficients which can introduce errors by representing changing in the filter bank energies. MFCC is calculated using equation 3.11

$$c(n) = \sum_{m=0}^{M-1}\big(s(m)\big)cos\left(\frac{\pi n(m-0.5)}{M}\right); \qquad n = 0,1,2,\dots\dots C-1 \qquad 3.11$$

$c(n)$ is the cepstral coefficients, and C is the number of MFCCs. Traditional MFCC systems use only 8–13 cepstral coefficients. The zeroth coefficient is often excluded since it represents the average log-energy of the input signal, which only carries little speaker-specific information.

Dynamic MFCC features: The cepstral coefficients are usually referred to as static features, since they only contain information from a given frame. The extra information about the temporal dynamics of the signal is obtained by computing first and second derivatives of cepstral coefficients. The first-order derivative is called delta coefficients, and the second-order derivative is called delta–delta coefficients. Delta coefficients tell about the speech rate, and delta–delta coefficients provide information similar to acceleration of audio Also known as differential and acceleration coefficients. The MFCC feature vector describes only the power spectral envelope of a single frame, but it seems like speech would also have information in the dynamics i.e. what are the trajectories of the MFCC coefficients over time. It turns out that calculating the MFCC trajectories and appending them to the original feature vector increases ASR performance by quite a bit (if we have 12 MFCC coefficients, we would also get 12 delta coefficients, which would combine to give a feature vector of length 24).

To calculate the delta coefficients, the following formula is used:

$$d_t = \frac{\sum_{n=1}^{N} n(\,c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \qquad 3.12$$

where d_t is a delta coefficient, from frame $t$computed in terms of the static coefficients c_{t+n} to c_{t-n}. A typical value for N is 2. Delta-Delta (Acceleration) coefficients are calculated in the same way, but they are calculated from the deltas, not the static coefficients.

3.5 Classification with Convolutional Neural Network

For the convolutional base we have used a common pattern one is a stack of Conv2D and second one is MaxPooling2D layers. Our model is sequential.

Table 3.1: Layer used in Convolution Neural Network, output shape of layer, Number of parameters used in the model

Layer (Type)	Output Shape	Param#
Conv2d(Conv2D)	(None,128,11,32)	320
Max_pooling2D(MaxPooling2D)	(None,64,6,32)	0
Batch_normalization_1(BatchNormalization)	(None,64,6,32)	128
Conv2D_1(Conv2D_1)	(None,62,4,32)	9248
Max_pooling2D_1(MaxPooling2D_1)	(None, 31 , 2, 32)	0
Batch_normalization_1(BatchNormalization_1)	(None,31,2,32)	128
Conv2d_2(Conv2D_2)	(None,30,1,32)	4128
Max_pooling2D_2(MaxPooling2D_2)	(Non,15,1,32)	0
Batch_normalization_2(BatchNormalization_2)	(None,15,1,32)	128
flatten (Flatten)	(None,480)	0
dense(Dense)	(None,64)	30784
dropout(Dropout)	(None,64)	0
dense_1(Dense	(None,10)	650

In our Model number of layers is fixed but number parameter changes as per the data experiments.

3.6 Activation function in Neural Network

A neural network activation function is a mathematical equation that determines the output of the built model. The function is attached to each neuron in the network, and determines whether it should be activated ("fired") or not, based on whether each neuron's input is relevant for the model's prediction. Activation functions also help normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

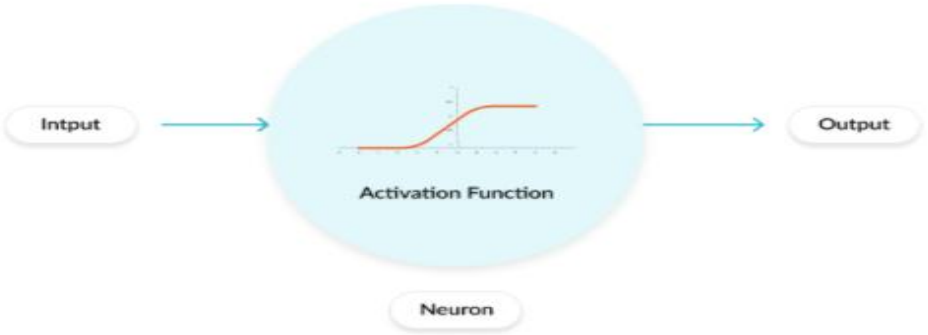


Fig 3.5 Activation Function of Neural Network

3.6.1 ReLU (Rectified Linear Unit)

- Computationally efficient—allows the network to converge very quickly
- Non-linear—although it looks like a linear function, ReLU has a derivative function and allows for backpropagation
- The Dying ReLU problem—when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.

3.6.2. Softmax

- Able to handle multiple classes only one class in other activation functions normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class.
- Useful for output neurons—typically Softmax is used only for the output layer, for neural networks that need to classify inputs into multiple categories.

Results and Discussion

Chapter-4 RESULTS AND DISCUSSION

This chapter includes the analysis of results obtained from the machine learning process. The whole process of music genre classification is divided into two main tasks. One is preprocessing and second one is implementing the CNN model for accuracy of the model. In the preprocessing we have organized the data set as per the compatibility of model and then we have extracted feature using MFCCs method and saved them into a JSON file.

4.1 Pre-Processing of Dataset

In this work we have used six semiclassical Indian music genre like Abhang, Bhajan, Kajari, Qawwali, Tappa and Thumri these are listed in table 4.1.

Table 4.1: Music Genre and Their class label

Genre	Class Label
Abhang	0
Bhajan	1
Kajari	2
Qawwali	3
Tappa	4
Thumri	5

The main challenge is that there is no existing data set available for Semi - Classical Indian music. The database is formed by collecting songs from various sites then we made it compatible for the model and converted each song to 30-second excerpts of .wav format.

Next task of the preprocessing is extracting the feature from this dataset. We have used the MFCCs technique for the feature extraction and saved all feature into a JSON file. In this part of preprocessing, we broke each song into 10 segments. We have used 450 songs for training the model and after preprocessing, we will have 4500 segments. Preprocessing and segmentation of data is shown in figure 4.1.

```
"D:\Thesis\Pycharm Project\venv\Scripts\python.exe" "D:/Thesis/Pycharm Project/preprocessing.py"

Processing: D:\Thesis\Music data base set\music Data_six_300\abhang
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:1
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:2
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:3
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:4
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:5
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:6
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:7
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:8
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:9
D:\Thesis\Music data base set\music Data_six_300\abhang\abhang001.wav, segment:10
```

Fig. 4.1: Preprocessing of Data Set

4.2 Implementing the Convolution Neural Network

Once the feature is extracted from the data then we can apply CNN for music genre classification. In this model we have used convolution layer to preserve the connection between pixels. Pooling layers is used in CNN to reduce the number of parameters when the images are too large. Max Pooling is used for discretization process and its is a down sampling method and it allow assumptions to be made about features contained in the sub-regions bin. Batch normalization is used for stabilizing the learning process and reducing the number of training epochs.

4.2.1 Result of The Model

CNN is a sequential model. Figure 4.2 show the layers used in CNN like Convolution layer , max pooling and dense.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 11, 32)       320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 6, 32)         0
_________________________________________________________________
batch_normalization (BatchNo (None, 64, 6, 32)         128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 62, 4, 32)         9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 2, 32)         0
_________________________________________________________________
batch_normalization_1 (Batch (None, 31, 2, 32)         128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 30, 1, 32)         4128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 1, 32)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 15, 1, 32)         128
_________________________________________________________________
flatten (Flatten)            (None, 480)               0
_________________________________________________________________
dense (Dense)                (None, 64)                30784
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
```

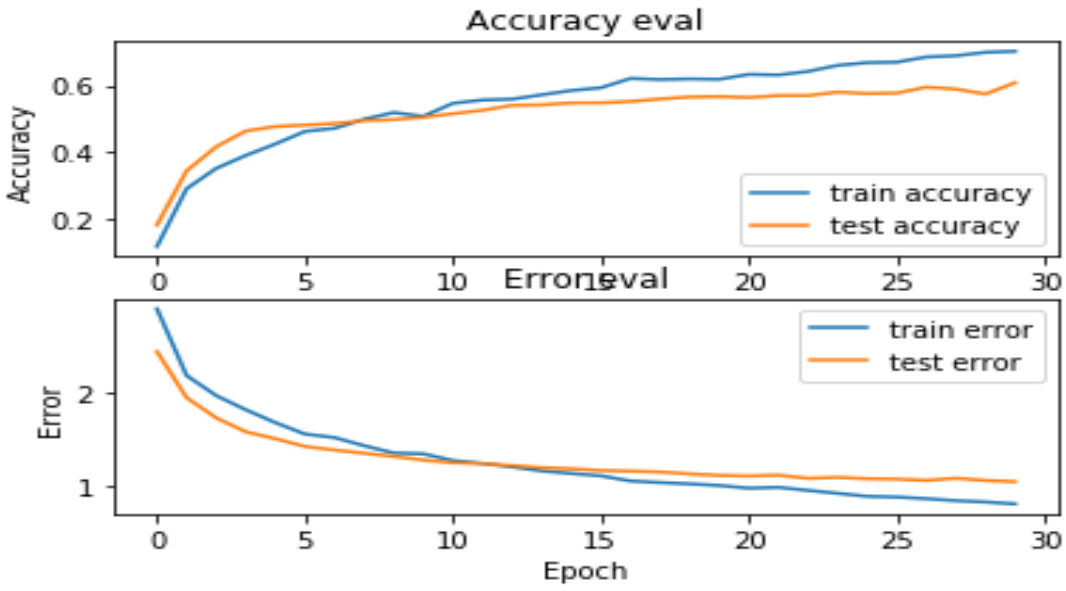
Fig 4.2: CNN Model

There are 2693 Data segment for training, 674 for validation and 1123 Data for testing. Figure 4.3 shows the accuracy and loss prediction of 30 Epoch.

```
Train on 2693 samples, validate on 674 samples
Epoch 1/30
2693/2693 [==============================] - 16s 6ms/sample - loss: 2.8853 - acc: 0.1151 - val_loss: 2.4320 - val_acc: 0.1795
Epoch 2/30
2693/2693 [==============================] - 15s 6ms/sample - loss: 2.1783 - acc: 0.2889 - val_loss: 1.9439 - val_acc: 0.3427
Epoch 3/30
2693/2693 [==============================] - 15s 5ms/sample - loss: 1.9652 - acc: 0.3502 - val_loss: 1.7294 - val_acc: 0.4154
Epoch 4/30
2693/2693 [==============================] - 15s 5ms/sample - loss: 1.8152 - acc: 0.3892 - val_loss: 1.5816 - val_acc: 0.4629
Epoch 5/30
2693/2693 [==============================] - 15s 6ms/sample - loss: 1.6789 - acc: 0.4237 - val_loss: 1.5079 - val_acc: 0.4763
Epoch 6/30
2693/2693 [==============================] - 12s 4ms/sample - loss: 1.5576 - acc: 0.4616 - val_loss: 1.4258 - val_acc: 0.4807
Epoch 7/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.5204 - acc: 0.4709 - val_loss: 1.3881 - val_acc: 0.4852
Epoch 8/30
2693/2693 [==============================] - 12s 4ms/sample - loss: 1.4340 - acc: 0.4983 - val_loss: 1.3544 - val_acc: 0.4941
Epoch 9/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.3566 - acc: 0.5184 - val_loss: 1.3200 - val_acc: 0.4970
Epoch 10/30
2693/2693 [==============================] - 12s 5ms/sample - loss: 1.3479 - acc: 0.5072 - val_loss: 1.2818 - val_acc: 0.5045
Epoch 11/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.2770 - acc: 0.5462 - val_loss: 1.2540 - val_acc: 0.5148
Epoch 12/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.2460 - acc: 0.5563 - val_loss: 1.2441 - val_acc: 0.5252
Epoch 13/30
2693/2693 [==============================] - 13s 5ms/sample - loss: 1.2092 - acc: 0.5589 - val_loss: 1.2229 - val_acc: 0.5401
Epoch 14/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.1667 - acc: 0.5719 - val_loss: 1.2012 - val_acc: 0.5415
Epoch 15/30
2693/2693 [==============================] - 11s 4ms/sample - loss: 1.1398 - acc: 0.5848 - val_loss: 1.1891 - val_acc: 0.5475
```

Fig 4.3: Accuracy and loss prediction of 30 Epoch

The test accuracy is 60% and train accuracy 79%. Figure 4.4 shows the test and train accuracy and error eval.




```
1123/1123 - 1s - loss: 1.0484 - acc: 0.6028

Test accuracy: 0.6028495
2693/2693 - 2s - loss: 0.6249 - acc: 0.7947

Train accuracy: 0.7946528
```

Fig. 4.4: Accuracy and Error eval of Model in 30 epochs

Figure 4.5 shows the confusion matrix of the model.

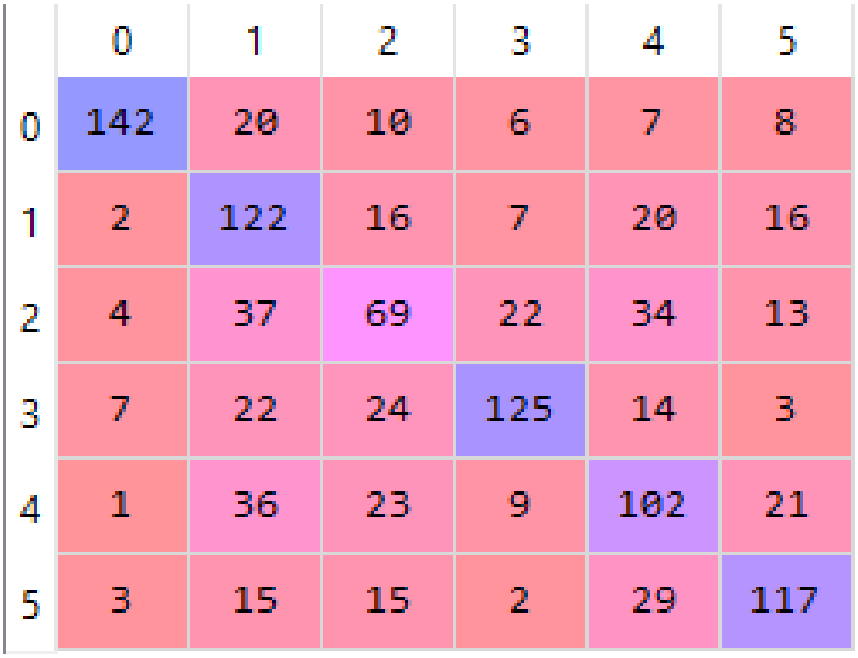


Fig. 4.5: Confusion Matrix of the model

4.3 Experiments of Model

4.3.1 Experiment 1

When we change the Testing Ratio. Environment setting for this experimental are listed in table 4.2.

Table 4.2: Test and Train accuracy when we change the testing ratio

Testing Ratio	Accuracy	
	Testing	Training
10%	63%	80%
20%	56%	79%
30%	60%	80%
40%	56%	81%

4.3.1.1 Experiment 1.1: When Testing Ratio is 10%

In the experiment 1.1, we have used 10% test data segment of the model. There are 3232 data segments for training, 449 data segments for test and 809 data segments for validation. Test accuracy of the 10 % testing ratio is 63.69% and train accuracy is 80.91%. The results of this experiments are shown in figure 4.6 and figure 4.7.

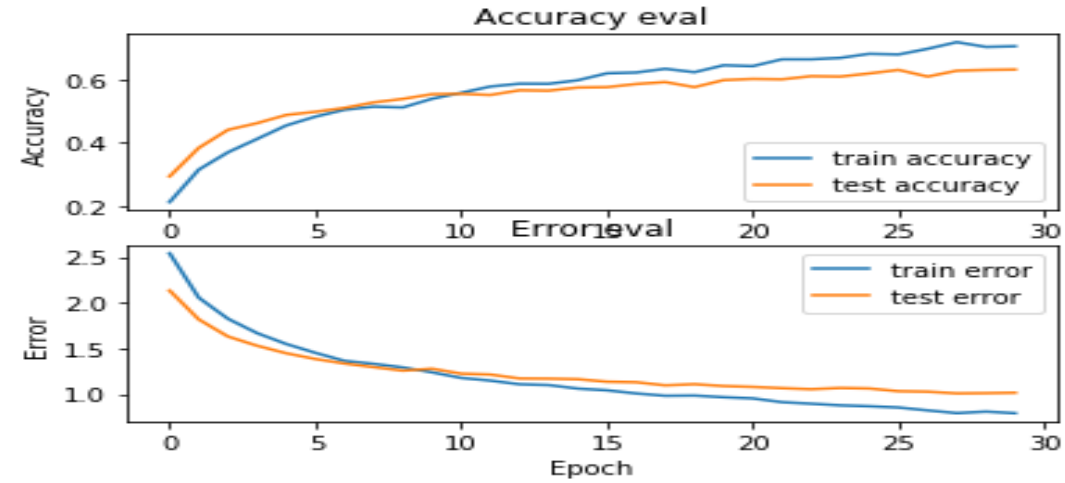


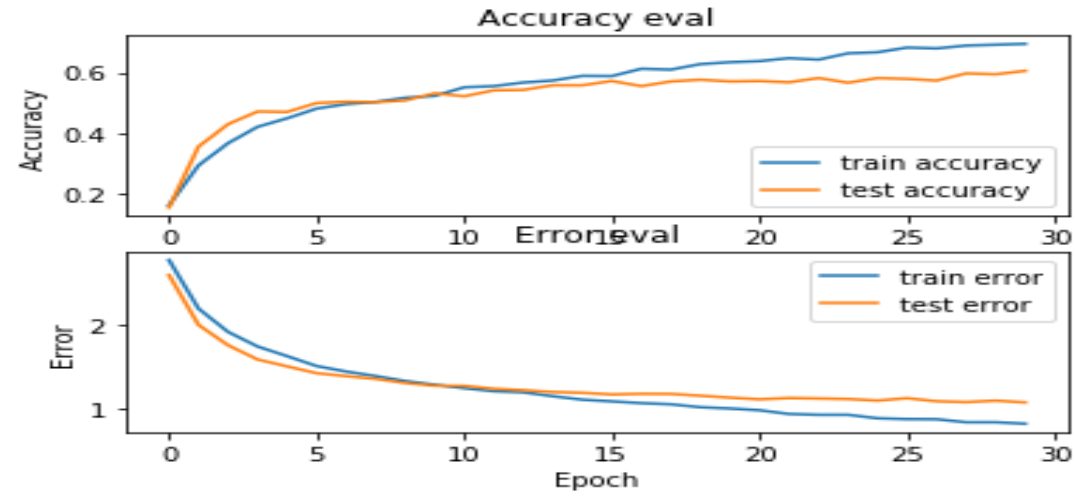
Fig.4.6: Accuracy and Error eval of Model when Testing ratio is 10%

	0	1	2	3	4	5
0	58	5	4	2	2	0
1	3	47	14	3	14	2
2	1	7	41	9	14	5
3	2	5	6	45	7	5
4	3	6	8	0	47	9
5	3	5	10	0	9	48

Fig.4.7 Confusion Matrix of the model when Testing ratio is 10%

4.3.1.2 Experiment 1.2: When Testing Ratio 20%

In the experiment 1.2, we have used 20% test data segment of the model. There are 2873 data segments for training, 898 data segments for test and 719 data segments for validation. Test accuracy of the 20 % testing ratio is 56.90% and train accuracy is 79.98%. The results of this experiments are shown in figure 4.8 and figure 4.9.




```
898/898 - 1s - loss: 1.0899 - acc: 0.5690

Test accuracy: 0.5690423
2873/2873 - 2s - loss: 0.6263 - acc: 0.7999

Train accuracy: 0.7998608
```

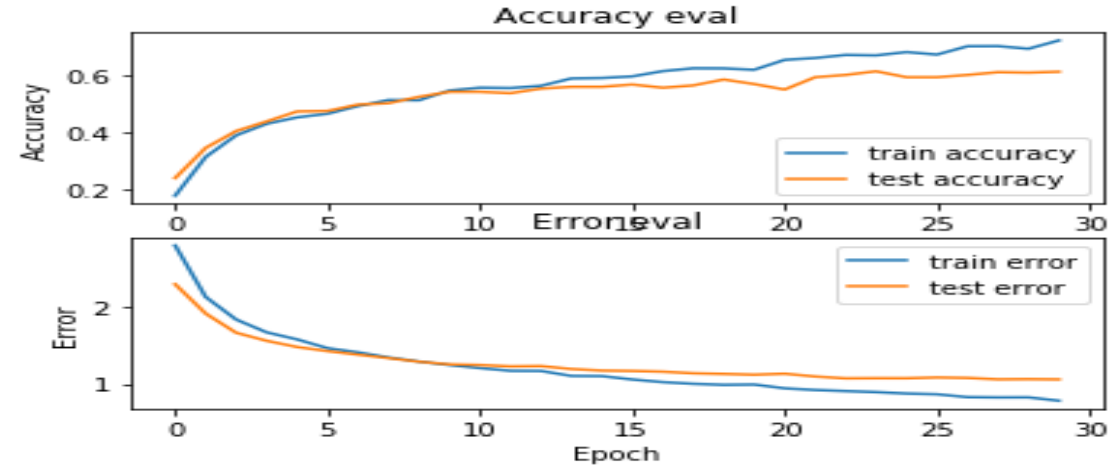
Fig 4.8: Accuracy and Error eval of Model when Testing ratio is 20%

	0	1	2	3	4	5
0	97	8	7	5	9	8
1	2	78	24	10	27	14
2	4	30	69	18	21	22
3	15	7	18	76	10	5
4	5	18	28	8	71	28
5	0	6	15	4	11	120

Fig. 4.9: Confusion Matrix of the model when Testing ratio is 20%

4.3.1.3 Experiment 1.3: When Testing Ratio 30%

In the experiment 1.3, we have used 30% test data segment of the model. There are 2514 data segments for training, 1347 data segments for test and 629 data segments for validation. Test accuracy of the 30 % testing ratio is 60.65% and train accuracy is 80.38%. The results of this experiments are shown in figure 4.10 and figure 4.11.




```
1347/1347 - 1s - loss: 1.0188 - acc: 0.6065

Test accuracy: 0.60653305
2514/2514 - 2s - loss: 0.5956 - acc: 0.8039

Train accuracy: 0.80389816
```

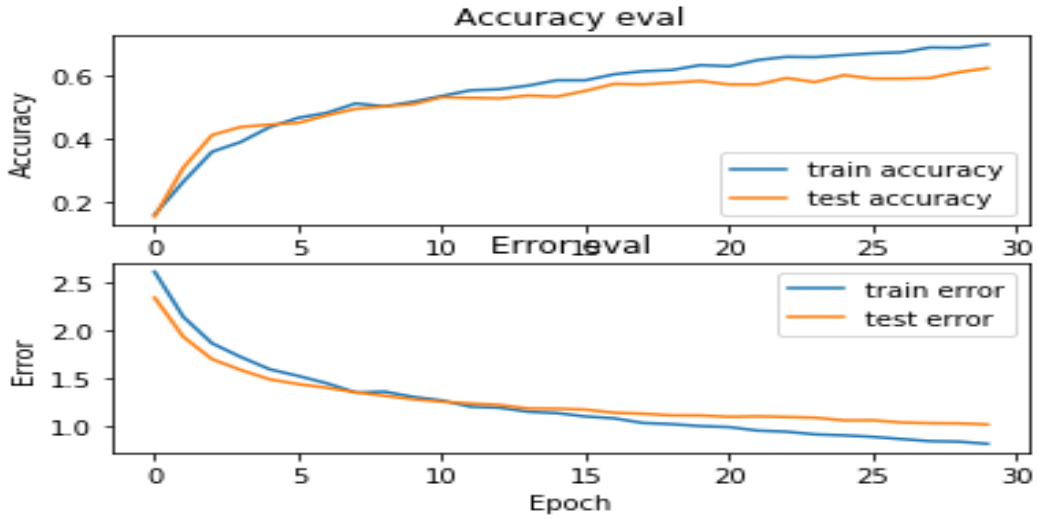
Fig 4.10: Accuracy and Error eval of Model when Testing ratio is 30%

	0	1	2	3	4	5
0	192	12	11	6	3	5
1	18	130	24	13	26	25
2	12	22	90	24	40	22
3	22	28	37	136	13	7
4	20	21	22	10	119	28
5	3	13	23	5	15	150

Fig. 4.11: Confusion Matrix of the model when Testing ratio is 30%

4.3.1.3 Experiment 1.4: When Testing Ratio 40%

In the experiment 1.4, we have used 40% test data segment of the model. There are 2155 data segments for training, 1796 data segments for test and 539 data segments for validation. Test accuracy of the 40 % testing ratio is 56.79% and train accuracy is 81.20%. The results of this experiments are shown in figure 4.12 and figure 4.13.




```
1796/1796 - 1s - loss: 1.1266 - acc: 0.5679

Test accuracy: 0.56792873
2155/2155 - 2s - loss: 0.6066 - acc: 0.8121

Train accuracy: 0.81206495
```

Fig 4.12: Accuracy and Error eval of model when testing ratio is 40%

	0	1	2	3	4	5
0	251	15	14	8	6	12
1	26	156	47	21	23	22
2	11	32	136	35	54	21
3	21	18	41	151	40	20
4	15	34	66	16	130	66
5	13	11	32	2	34	196

Fig. 4.13: Confusion Matrix of the model when testing ratio is 40%

4.3.2 Experiment 2: Combined music genre

In the experiment 2 we have combined the two genres and then trained the model for better performance and test the accuracy of model. Environmental setting for this experiments are listed in table 4.3

Table 4.3: Test and Train accuracy when we combine two music genre

Genre	Accuracy	
	Testing	**Training**
Genre 4(Abhang+Bhajan, Tappa+Kajari)	66%	81%
Genre 5(Abhang+Bhajan)	60%	78%

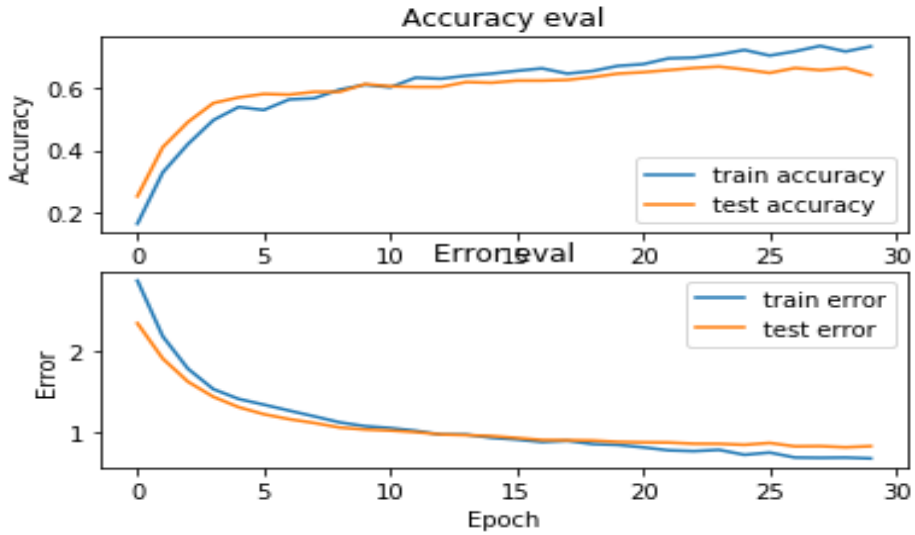
4.3.2.1 Experiment 2.1

In this experiment we have combined the Abhang and bhajan into one genre and tappa or kajari into one genre. Environmental setting for this experiments are listed in table 4.4 The results of this experiments are shown in figure 4.14 and figure 4.15.

Table 4.4: Music genre and their class label when we combined Abhang and Bhajan, and Kajari and Tappa

Genre	Class Lebel
Abhang+ Bhajan	0
Kajari+ Tappa	1
Qawwali	2
Thumri	3

In this experiment we have used 1793 samples for training, 449 samples for validation and 748 samples for testing. The testing accuracy is 66.04% and training accuracy is 81.31%.




```
748/748 - 1s - loss: 0.8885 - acc: 0.6604

Test accuracy: 0.6604278
1793/1793 - 1s - loss: 0.5102 - acc: 0.8132

Train accuracy: 0.8131623
```

Fig 4.14 Accuracy and Error eval of Model when Abhang and Bhajan, and Tappa and Kajari is combined

	0	1	2	3
0	138	22	22	11
1	30	88	25	38
2	31	21	137	3
3	11	33	7	131

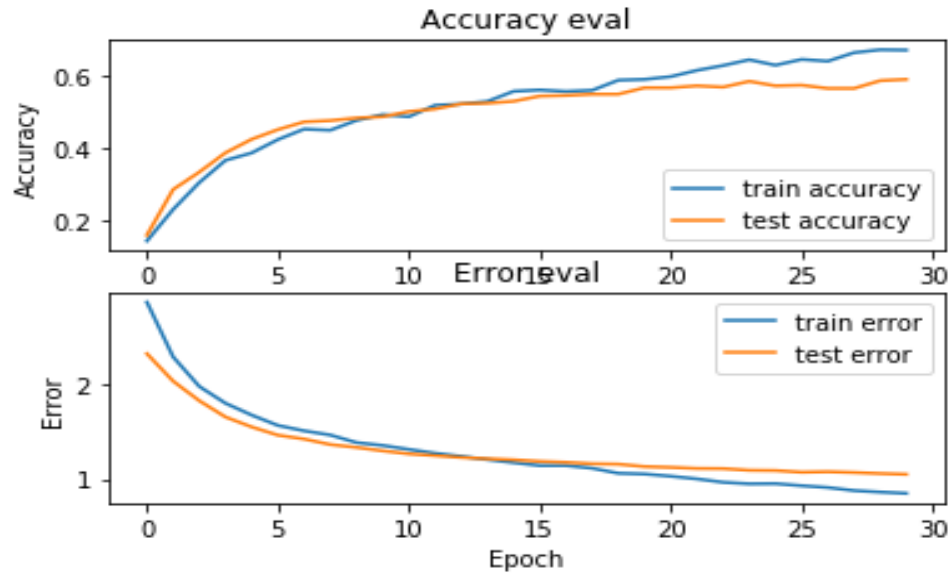
Fig. 4.15: Confusion Matrix of Model when Genre (Abhang+Bhajan, Tappa+Kajari)

4.3.4.2 Experiment 2.2

In this experiment we have combined the Abhang and Bhajan into one genre. We have used 2244 data samples for training, 561 for validation and 935 for testing the accuracy of model. Environmental setting for this experiments are listed in table 4.5. The results of this experiments are shown in figure 4.16 and figure 4.17.

Table 4.5: Music genre and their class label when we combine Abhang and Bhajan Testing accuracy is 60.11% and train accuracy is 78.92%.

Genre	Class Lebel
Abhang + Bhajan	0
Kajari	1
Qawwali	2
Tappa	3
Thumri	4




```
935/935 - 1s - loss: 0.9950 - acc: 0.6011

Test accuracy: 0.6010695
2244/2244 - 2s - loss: 0.6555 - acc: 0.7892

Train accuracy: 0.7892157
```

Fig 4.16: Accuracy and Error eval of Model when Abhang and Bhajan is combined

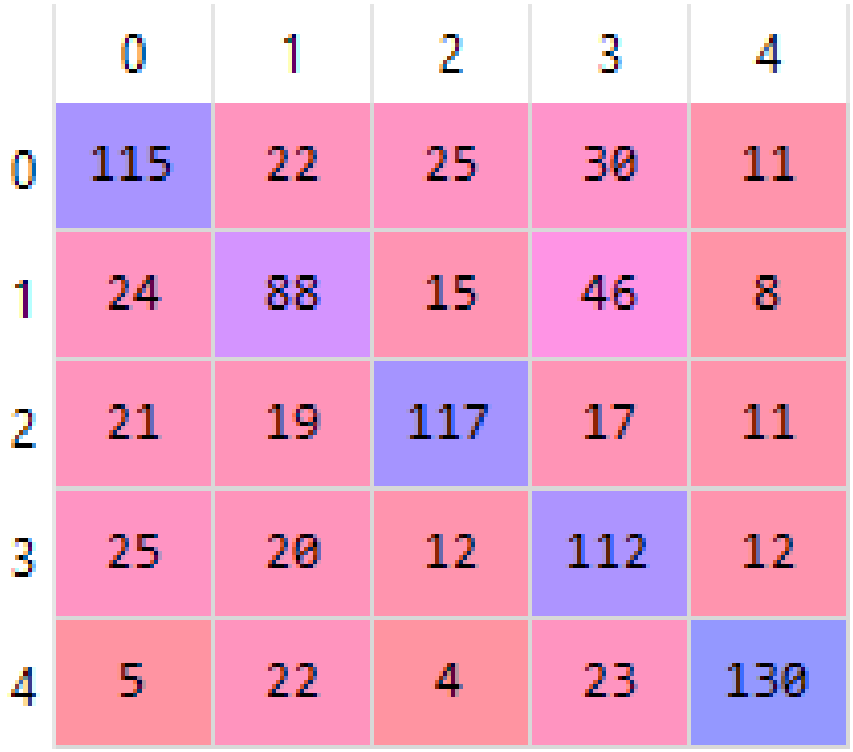


Fig 4.17: Confusion Matrix of Model when Genre Abhang and Bhajan is combined

4.3.3. Experiment 3 When we change the number of Data set.

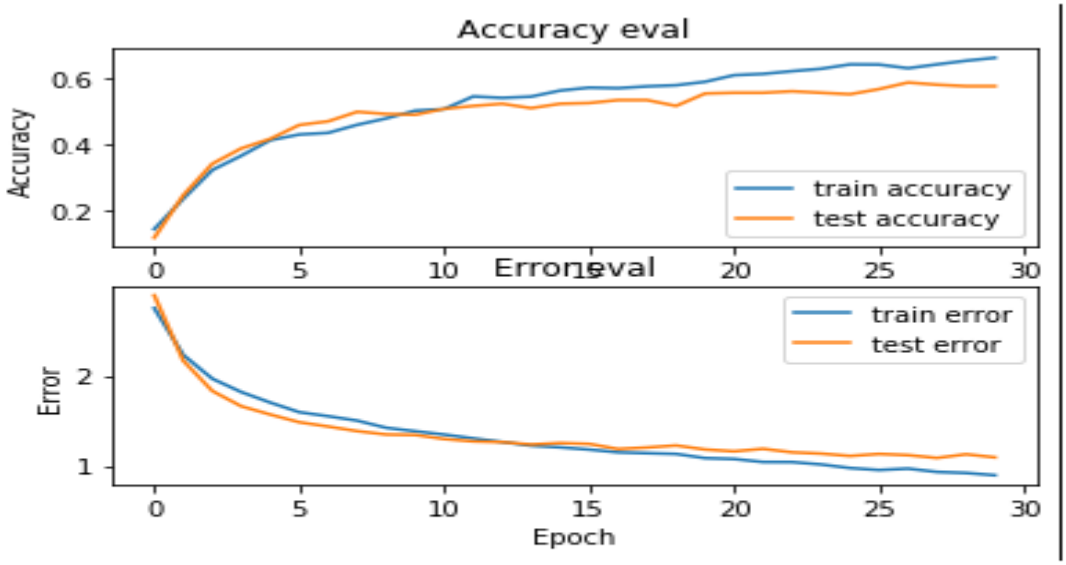
Environmental setting for this experiments are listed in table 4.6.

Table 4.6: Train and Test accuracy when we change the number of data set

Number of Song in Data set	Accuracy	
	Testing	**Training**
300	56%	78%
450	62%	79%

4.3.3.1 Experiment 3.1: When Data set is 300

In experiment 3.1, we have used six genres and each genre have 50 data song. When we extract genre's feature it will convert 3000 data sample. The test accuracy of the 300 data set is 56.93% and train accuracy is 78.17%. The results of this experiments are shown in figure 4.18 and figure 4.19.




```
750/750 - 0s - loss: 1.1465 - acc: 0.5693

Test accuracy: 0.5693333
1800/1800 - 1s - loss: 0.6888 - acc: 0.7817

Train accuracy: 0.7816667
```

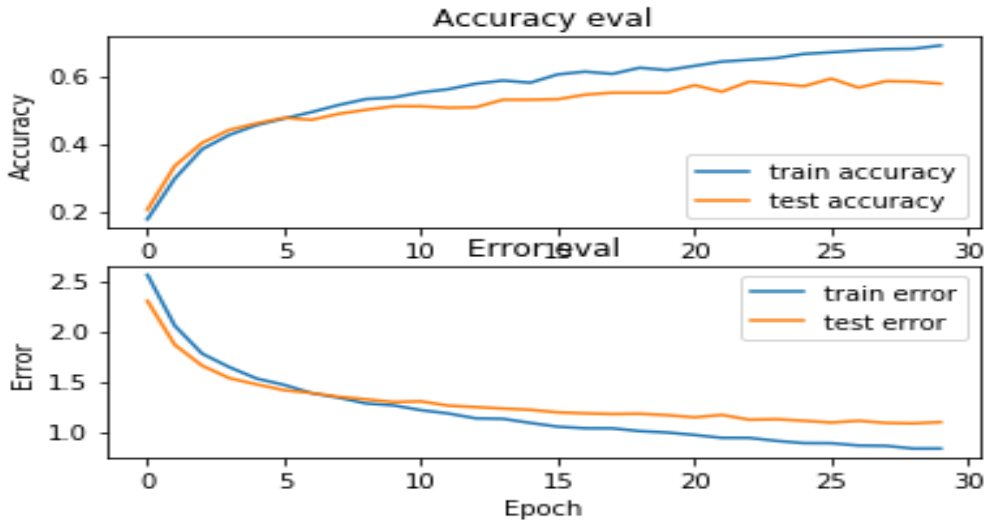
Fig 4.18: Accuracy and Error eval of 300 songs

	0	1	2	3	4	5
0	79	10	11	2	3	4
1	10	70	8	9	21	11
2	4	18	62	11	20	19
3	15	9	16	73	3	9
4	6	15	25	5	56	23
5	0	6	11	0	19	87

Fig 4.19: Confusion matrix of the 300 songs

4.3.3.2 Experiment 3.2: When Data set is 450

In this experiment, we have used six genres and each genre have 75 data song. When we extract genre's feature it will co create 4500 segmentation of data sample. The testing accuracy for 450 data set is 62.15 % and training accuracy is 78.53%. Results of this experiment are shown in figure 4.20 and figure 4.21




```
1123/1123 - 1s - loss: 1.0594 - acc: 0.6215

Test accuracy: 0.6215494
2693/2693 - 2s - loss: 0.6433 - acc: 0.7854

Train accuracy: 0.78536946
```

Fig 4.20: Accuracy and Error eval of 450 songs

	0	1	2	3	4	5
0	153	7	8	8	11	5
1	4	104	34	15	27	11
2	6	16	98	22	42	11
3	20	9	16	126	24	2
4	11	11	25	12	99	7
5	10	8	20	1	22	118

Fig 4.21: Confusion matrix of the 450 songs

4.3.4. Experiment 4: Comparison with Western Data

In experiment 4 we have calculated the testing and training accuracy of the western data set. The testing accuracy of western data is 69.02% and train accuracy is 83.31%. The results of this experiments are shown in figure 4.22.

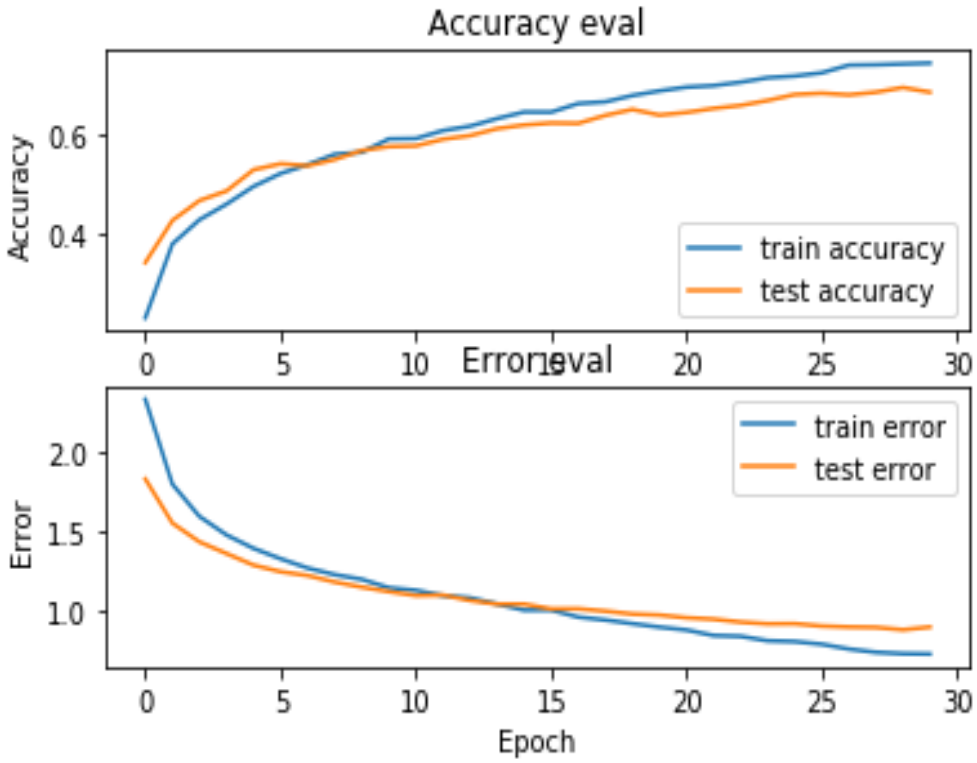


Fig 4.22: Accuracy and Error eval of western dataset

4.4 Discussion

The implemented model helps to accurate the accuracy of the semi classical data set which is the test accuracy is 60% and train accuracy 79%. In this model we change the testing ratio and got the maximum train accuracy when we used 40% of data sample for testing. The second experiment we combined the genre Abhang or bhajan and Tappa or kajari we got the test accuracy 66% and train accuracy 81%.

Summary and Conclusion

Chapter-5 SUMMARY AND CONCLUSION

5.1 SUMMARY

In this work we have tried to understand the Indian music and their genre classification by using the MFCC features extracting technique. Developed model show various experiments in this study. The model helps us to show the test and train accuracy.

The main motive of this thesis work is to explore the areas of genre recognition, classification because as compare to western music less work has been done in Indian music.

5.2 CONCLUSION

The overall conclusion of the model shows that:

1. Study MFCC feature extracting technique.
2. Accuracy of the train model up to 79%.
3. Implementation of neural network and doing various experiment like changing the number of data set, change the test ratio, combined the music genre and compare with western data set.
4. The maximum train accuracy of the changing number of datasets is 81%.
5. The train accuracy is 81% when we combine music genre like Abhang or Bhajan into one genre and Tappa or Kajari into another genre. The final genre combination is four.
6. The train accuracy is 78% when we combine music genre like Abhang or Bhajan into one genre. The final genre combination is five.
7. The train accuracy is 78% when we used 300 data song.

5.3 Future Scope

There are several machine leaning techniques for the classification and feature extraction to explore and analyzes the music. In this work we have classified the Indian

semi classical music. Indian music is very vast and lot of work could be done in future like Mood classification, Music Recommendation, Artist identification, Artist similarity, cover song detection, Rhythm and beat detection, Score following like Chord detection, Organization of music, Audio Fingerprinting, Audio segmentation, Instrument detection, Automatic source, separation like Onset detection, Optical music recognition, Melody transcription.

Literature Cited

LITERATURE CITED

A. Samad, S. and B. Huddin, A., 2018. Genre Classification of Traditional Malay Music Using Spectrogram Correlation. International Journal of Engineering & Technology, 7(4.11), p.29.

Anglade, A., Benetos, E., Mauch, M. and Dixon, S., 2010. Improving Music Genre Classification Using Automatically Induced Harmony Rules. Journal of New Music Research, 39(4), pp.349-361.

Arpit Seth, 2020. Genre prediction for music recommendation using machine learning. EPRA International Journal of Research & Development (IJRD), pp.206-210.

Betsy. S and Bhalke. D. G, 2015. Genre Classification of Indian Tamil Music using Mel-Frequency Cepstral Coefficients. International Journal of Engineering Research and, V4(12).

Bhalke, D., Rajesh, B. and Bormane, D., 2017. Automatic Genre Classification Using Fractional Fourier Transform Based Mel Frequency Cepstral Coefficient and Timbral Features. Archives of Acoustics, 42(2), pp.213-222.

Dabas, C., Agarwal, A., Gupta, N., Jain, V. and Pathak, S., 2020. Machine Learning Evaluation for Music Genre Classification of Audio Signals. International Journal of Grid and High-Performance Computing, 12(3), pp.57-67.

Davis, S. Mermelstein, P., 1980 Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. In IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357-366.

Elbir, A. and Aydin, N., 2020. Music genre classification and music recommendation by using deep learning. Electronics Letters, 56(12), pp.627-629.

G.K.T. Ganchev, N. Fakotakis, **2005** Comparative evaluation of various MFCC implementations on the speaker verification task, in Proceedings of International Conference on Speech and Computer (SPECOM), pp.191–194.

J. Benesty, M.M. Sondhi, Y.A. Huang, **2008** Handbook of Speech Processing (Springer, New York).

J. Volkmann, S. Stevens, E. Newman, **1937** A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. **8**, 185–190.

J. S. Mason, X. Zhang, **1991** Velocity and acceleration features in speaker recognition, in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (1991), pp. 367-3676.

J.W. Picone, **1993** Signal modeling techniques in speech recognition. Proc. IEEE **81**, 1215–1247.

J.R. Deller, J.H. Hansen, J.G. Proakis, **1993** Discrete Time Processing of Speech Signals (Prentice Hall, NJ).

Kao, M., Yang, C. and Shiau, S., 2009. Tempo and beat tracking for audio signals with music genre classification. International Journal of Intelligent Information and Database Systems, 3(3), p.275.

L. Rabiner, B.-H. Juang, B. Yegnanarayana, **2008** Fundamentals of Speech Recognition (Pearson Education, London).

Li, W., Zhang, X. and Wang, Z., 2013. Music content authentication based on beat segmentation and fuzzy classification. EURASIP Journal on Audio, Speech, and Music Processing, 2013(1).

Marques, G., Langlois, T., Gouyon, F., Lopes, M. and Sordo, M., 2011. Short-term Feature Space and Music Genre Classification. Journal of New Music Research, 40(2), pp.127-137.

Meenakshi K, SafaM., Geetha G., Saranya G., Sundara Kanchana J, 2008 International Journal of Recent Technology and Engineering, Music Genre Classification using Lyric Mining Based ontf-Idf. 8(5), pp.36-40.

Mutiara, A., Refianti, R. and Mukarromah, N., 2016. Musical Genre Classification Using SVM and Audio Features. TELKOMNIKA (Telecommunication Computing Electronics and Control), 14(3), p.1024.

Ntalampiras, S., 2014. Directed Acyclic Graphs for Content Based Sound, Musical Genre, and Speech Emotion Classification. Journal of New Music Research, 43(2), pp.173-182.

Rajesh, B. and Bhalke, D., 2016. Automatic genre classification of Indian Tamil and western music using fractional MFCC. International Journal of Speech Technology, 19(3), pp.551-563.

S. Furui, **1981** Comparison of speaker recognition methods using statistical features and dynamic features. IEEE Trans. Acoust. Speech Sig. Proc. **29**, 342–350.

Sanden, C., Befus, C. and Zhang, J., 2012. A Perceptual Study on Music Segmentation and Genre Classification. Journal of New Music Research, 41(3), pp.277-293.

Skillman, T., 1986. The Bombay Hindi Film Song Genre: A Historical Survey. Yearbook for Traditional Music, 18, p.133.

Song, Y. and Zhang, C., 2008. Content-Based Information Fusion for Semi-Supervised Music Genre Classification. IEEE Transactions on Multimedia, 10(1), pp.145-152.

Tzanetakis, G. and Cook, P., 2002. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), pp.293-302.

Vishnupriya, S. and Meenakshi, K., 2018. Automatic Music Genre Classification using Convolution Neural Network. 2018 International Conference on Computer Communication and Informatics (ICCCI).

Viswanathan, A., 2016. Music Genre Classification. International Journal Of Engineering And Computer Science.

X. Huang, A. Acero, and H. Hon. **2001** Spoken Language Processing: A guide to theory, algorithm, and system development. Prentice Hall.

Yoon, J., Lim, H. and Kim, D., 2016. Music Genre Classification Using Feature Subset Search. International Journal of Machine Learning and Computing, 6(2), pp.134-138.

Yu, B. and Xu, Z., 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowledge-Based Systems, 21(4), pp.355-362.

Z. Fang, Z. Guoliang, S. Zhanjiang, **2000** Comparison of different implementations of MFCC. J. Comput. Sci.Technol.**16**,582–589.

Appendices

Screenshot of some part of code that has been used in proposed work.

1. Mapping with Music genre and class label

```
def save_mfcc(dataset_path, json_path, num_mfcc=13, n_fft=2048, hop_length=512, num_segments=5):

    data = {
        "mapping": [],
        "labels": [],
        "mfcc": []
    }

    samples_per_segment = int(SAMPLES_PER_TRACK / num_segments)
    num_mfcc_vectors_per_segment = math.ceil(samples_per_segment / hop_length)


    for i, (dirpath, dirnames, filenames) in enumerate(os.walk(dataset_path)):


        if dirpath is not dataset_path:


            semantic_label = dirpath.split("/")[-1]
            data["mapping"].append(semantic_label)
            print("\nProcessing: {}".format(semantic_label))
```

2. MFCC code for feature extraction

```
for f in filenames:


    file_path = os.path.join(dirpath, f)
    signal, sample_rate = librosa.load(file_path, sr=SAMPLE_RATE)

    for d in range(num_segments):


        start = samples_per_segment * d
        finish = start + samples_per_segment


        mfcc = librosa.feature.mfcc(signal[start:finish], sample_rate, n_mfcc=num_mfcc, n_fft=n_fft,
                                    hop_length=hop_length)
        mfcc = mfcc.T


        if len(mfcc) == num_mfcc_vectors_per_segment:
            data["mfcc"].append(mfcc.tolist())
            data["labels"].append(i - 1)
            print("{}, segment:{}".format(file_path, d + 1))
```

3. Code for Convolution Neural Network Model

```
def build_model(input_shape):

    model = keras.Sequential()


    model.add(keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))
    model.add(keras.layers.BatchNormalization())


    model.add(keras.layers.Conv2D(32, (3, 3), activation='relu'))
    model.add(keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'))
    model.add(keras.layers.BatchNormalization())


    model.add(keras.layers.Conv2D(32, (2, 2), activation='relu'))
    model.add(keras.layers.MaxPooling2D((2, 2), strides=(2, 2), padding='same'))
    model.add(keras.layers.BatchNormalization())

    model.add(keras.layers.Flatten())
    model.add(keras.layers.Dense(64, activation='relu'))
    model.add(keras.layers.Dropout(0.3))


    model.add(keras.layers.Dense(10, activation='softmax'))

    return model
```

4. Code for Model prediction

```
def predict(model, X, y):
    optimiser = keras.optimizers.Adam(learning_rate=0.0001)
    model.compile(optimizer=optimiser,  loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    model.summary()
    history = model.fit(X_train, y_train, validation_data=(X_validation, y_validation), batch_size=32, epochs=30)

    plot_history(history)


    test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)
    print('\nTest accuracy:', test_acc)
    X_to_predict = X_test[100]
    y_to_predict = y_test[100]

    predict(model, X_to_predict, y_to_predict)
```

Vita

Neema Bhandari, *author of this manuscript was born on 15th May 1990. She is a resident of Bhimtal (Uttarakhand). She completed her High-school in 2005 and Intermediate in 2007 from Government Inter Collage Kaghthariya, Haldwani and Diploma in 2010 from Govt. Polytechnic, Dwarahat. She did his graduation from Amrapali institute of technology and sciences, Haldwani, Uttarakhand Technical University, Dehradun in 2013. She took admission for the degree of M.Tech., (Computer Engineering) in College of College of Technology, G.B. Pant University of Agriculture and Technology, Pantnagar in the year 2018.*

Permanent Address

Ms. Neema Bhandari
D/O Ram Singh
Village- Paniyali, Post-Kaghthariya
Dist. Nainital
Uttarakhand
Email : neemabhandari22@gmail.com

ABSTRACT

Name	:	Neema Bhandari	**Id. No.**	:	54100
Sem. and year of admission	:	1st Sem., 2018-19	**Degree**	:	M.Tech.
			Department	:	Computer Engineering
Major	:	Computer Engineering			
Thesis title	:	"Music Genre Classification using Convolutional Neural Network"			

Advisor : Prof. B. K. Singh

Music applications are one of the most used applications in the world. Music Genre Classification (MGC) has been gaining attention with the rise of digital music and it is a useful tool for semantic information to music tracks in offline and online music collections. A music genre refers to a specific class of music with a set of common properties. A mere perception of the music of that class can help one to distinguish it from other classes. Musical genre classification is a promising yet difficult task in the field of musical information retrieval. To determine the genre of a song it has to be distinguished by its unique audio features so that its contents can be analyzed with respect to the produced wave signals. A single Music Genre is a set of different features that combined with a specific pattern of rhythm, melody, harmony, instruments, mood and attitude, lyrics and language. In the Western music there has been much work done in the area of automatic tagging genre recognition, classification and comparative studies as compare to the Indian music. As a widely used feature in genre classification systems, Mel-frequency cepstral coefficients (MFCC) is typically believed to encode timbral information, since it represents short-duration musical textures. In this thesis, we investigate the invariance of MFCC and show that MFCCs in fact encode both timbral and key information. Convolutional Neural Networks have additional layers for edge detection that make them well suited for classification problems. For the convolutional base classification, we used a common pattern one is a stack of Conv2D and second one is MaxPooling2D layers. In this research work we use six semi classical Indian music genre like Abhang, Bhajan, Kajari, Qawwali, Tappa and Thumri.

(B. K. Singh)
Advisor

(Neema Bhandari)
Author

सारांश

नाम	:	**नीमा भंडारी**	**परिचयांक**	:	५४१००
प्रवेश का सत्र एवं वर्ष	:	प्रथम षट्मास २०१८-१९	**उपाधि**	:	स्नातकोत्तर अभियांत्रिकी
मुख्य विषय शोध	:	संगणक अभियांत्रिकी	**विभाग**	:	संगणक अभियांत्रिकी
शोध ग्रन्थ का शीर्षक	:	**“कन्वेंशन न्यूरल नेटवर्क का उपयोग करके संगीत शैली का वर्गीकरण”**			
सलाहकार	:	प्राध्यापक बी . के. सिंह			

संगीत अनुप्रयोग दुनिया में सबसे अधिक उपयोग किए जाने वाले अनुप्रयोगों में से एक है। संगीत शैली का वर्गीकरण (MGC) डिजिटल संगीत के उदय के साथ ध्यान आकर्षित कर रहा है और यह ऑफ़लाइन और ऑनलाइन संगीत संग्रह में संगीत की जानकारी के लिए एक उपयोगी उपकरण है। एक संगीत शैली सामान्य गुणों के एक सेट के साथ संगीत के एक विशिष्ट वर्ग को संदर्भित करती है। उस वर्ग के संगीत की एक मात्र धारणा इसे अन्य वर्गों से अलग करने में मदद कर सकती है। संगीत जानकारी पुनर्प्राप्ति के क्षेत्र में संगीत शैली का वर्गीकरण एक आशाजनक लेकिन कठिन काम है। एक गीत की शैली को निर्धारित करने के लिए इसकी अनूठी ऑडियो विशेषताओं द्वारा प्रतिष्ठित किया जाना है ताकि इसकी सामग्री का उत्पादन तरंग संकेतों के संबंध में विश्लेषण किया जा सके। एक एकल संगीत शैली अलग-अलग विशेषताओं का एक सेट है जो लय, माधुर्य, सद्भाव, वाद्य, मूड और दृष्टिकोण, गीत और भाषा के एक विशिष्ट पैटर्न के साथ संयुक्त है। पश्चिमी संगीत में, भारतीय संगीत की तुलना में स्वत: टैगिंग शैली मान्यता, वर्गीकरण और तुलनात्मक अध्ययन के क्षेत्र में बहुत काम किया गया है। शैली वर्गीकरण प्रणालियों में व्यापक रूप से इस्तेमाल की जाने वाली सुविधा के रूप में, मेल-फ्रीक्वेंसी सेफस्ट्राल गुणांक (एमएफसीसी) को आमतौर पर टाइमब्रल जानकारी को एन्कोड करने के लिए माना जाता है, क्योंकि यह छोटी अवधि के संगीत बनावट का प्रतिनिधित्व करता है। इस थीसिस में, हम एमएफसीसी के आक्रमण की जांच करते हैं और दिखाते हैं कि एमएफसीसी वास्तव में टाइमब्रल और महत्वपूर्ण जानकारी दोनों को एनकोड करते हैं। संवादात्मक तंत्रिका नेटवर्क में किनारे की पहचान के लिए अतिरिक्त परतें हैं जो उन्हें वर्गीकरण समस्याओं के लिए अच्छी तरह से अनुकूल बनाती हैं। इस शोध कार्य में हम छह अर्ध शास्त्रीय भारतीय संगीत शैली जैसे अभंग, भजन, कजरी, कव्वाली, टप्पा और ठुमरी का उपयोग करते हैं.

(बी . के. सिंह)
सलाहकार

(नीमा भंडारी)
शोधकर्ता